Algorithm 54

APPROXIMATION OF STRAIGHT LINES

A. H. J. Sale
Basser Computing Dept.
University of Sydney

Author's note:

This routine implements a method described by J. E. Bresenham (1965) for the control of an incremental digital plotter in USASI FORTRAN. A similar algorithm has been published earlier (Stockton, 1963) which is externally identical to that given here (it produces the same set of increments): this version is simpler and is written to achieve fast execution of the inner loop. No multiplications or divisions are needed.

Consider a Cartesian grid of lines of unit spacing, and a plotter which may plot points only at intersections of the grid lines. Then given two points (x_1, y_1) and (x_2, y_2) on the grid

the routine supplies a set of increments $(\Delta x_i, \Delta y_i)$ which when added successively to (x_1, y_1) produce an approximation to the straight line joining these two points, and such that no increment is greater than 1 in magnitude. The minimum number of increments is obviously

$$\max(\text{abs}(x_1 - x_2), \text{abs}(y_1 - y_2))$$

and in fact this number are supplied.

It can be seen from the above that of the two increments Δx_i and Δy_i, one will always be unity (in magnitude) after a call to the routine, and therefore more compact ways of returning the results than that chosen are possible. However since the purpose of the algorithm is to present a method, and since in most applications it will probably be rewritten in a suitable assembly language or recast in form, it was thought unwise to obscure the basic technique.

To illustrate the choice of points by the algorithm a 'print-plot' produced by a sample driver program is shown in Fig. 1. This consists of the numeral '2' (plotted as five straight-line segments) with various scales and in various orientations.

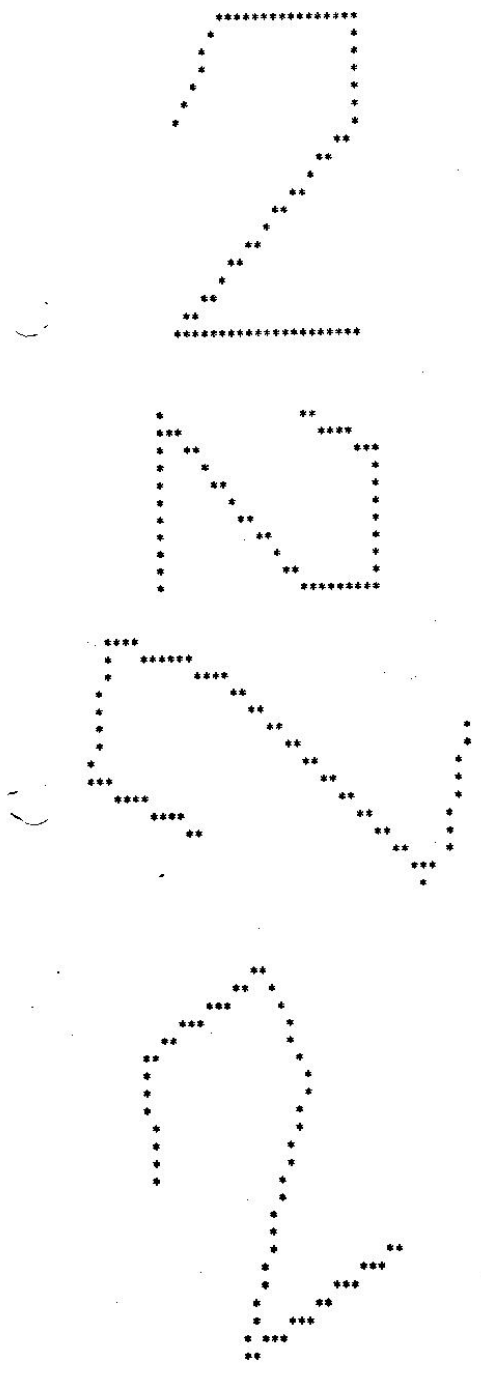

Fig. 1. Points chosen by LLINE

```
      SUBROUTINE LLINE(KX1,KY1,KX2,KY2,KXRAY,KYRAY,MAX,LENG)
C
C     LLINE PRODUCES AN APPROXIMATION TO THE STRAIGHT LINE BETWEEN THE
C     POINTS (KX1,KY1) AND (KX2,KY2), USING ONLY UNIT INCREMENTS
C     IN THE X-DIRECTION, IN THE Y-DIRECTION, OR BOTH TOGETHER.
C
C     THE (X,Y) INCREMENTS FOR EACH STEP ARE STORED IN THE ARRAYS
C     KXRAY AND KYRAY, WHICH MUST BE DECLARED IN THE CALLING
C     PROGRAM AS ONE-DIMENSIONAL AND AT LEAST OF LENGTH MAX.
C     IT IS SIMPLE TO MAKE THE ARRAYS STORE INSTEAD THE COORDINATES
C     WHICH CAN BE COMPUTED INSIDE THE DO LOOP.
C
C     THE VARIABLE LENG IS SET TO INDICATE THE LENGTH OF THE USED
C     PORTION OF THE ARRAYS. IF THIS IS EQUAL TO MAX THE ROUTINE
C     MAY HAVE RUN OUT OF ARRAY SPACE. TO PREVENT THIS, MAX MUST
C     BE AT LEAST EQUAL TO THE LARGEST (IN MAGNITUDE) OF
C     KX2-KX1 AND KY2-KY1.
C
C     SPECIFICATIONS
      INTEGER KX1,KY1,KX2,KY2,MAX,LENG
      INTEGER KXRAY(MAX),KYRAY(MAX)
C     DECLARATIONS
      INTEGER KDELA,KDELB,KDELI,KDELB2,KDELD2
      INTEGER INCX1,INCX2,INCY1,INCY2
      INTEGER J
C     EQUIV NOT NEEDED, JUST IN TO SAVE SPACE, AND FOR MNEMONICS
      EQUIVALENCE (KDELB,KDELB2)
C
C     SET UP THE INITIAL POINT COORDINATES, DELTA X AND DELTA Y,
C     AND COMPUTE THE 45 DEGREE INCREMENTS
      CALL SGNABS(KX2-KX1,INCX2,KDELA)
      CALL SGNABS(KY2-KY1,INCY2,KDELB)
C
C     NOW TEST THE MAGNITUDES OF DELTA X AND DELTA Y TO CHOOSE THE
C     APPROPRIATE COORDINATE SYSTEM A AND B. ALSO GET THE INCREMENTS
C     ALONG THE AXES AS REQUIRED
      IF (KDELA.LT.KDELB) GOTO 1
      INCX1=INCX2
      INCY1=0
      GOTO 2
    1 INCX1=KDELA
      KDELA=KDELB
      KDELB=INCX1
      INCX1=0
      INCY1=INCY2
C
C     NOW SET UP TWO TIMES DELTA A AND B, AND INITIAL DELTA I
    2 KDELB2=KDELB*KDELB
      KDELD2=KDELA*KDELA-KDELB2
      KDELI=KDELD2-KDELA
C
C     GO IN A LOOP FOR MAX TIMES OR UNTIL REACHING KDELA
      DO 4 J=1,MAX
C
C     TEST FOR DELTA A ZERO OR ONE, OR THE LINE FINISHED
      IF (J.GT.KDELA) GOTO 5
C     TEST THE SIGN OF DELTA I TO CHOOSE THE APPROPRIATE INCREMENTS
C     THEN COMPUTE NEXT POINT AND NEW DELTA I
      IF (KDELI.GE.0) GOTO 3
      KXRAY(J)=INCX1
      KYRAY(J)=INCY1
      KDELI=KDELI+KDELB2
      GOTO 4
    3 KXRAY(J)=INCX2
      KYRAY(J)=INCY2
      KDELI=KDELI-KDELD2
    4 CONTINUE
C
C     RUN OUT OF ARRAY SPACE
C     THE AVAILABLE MAX ELEMENTS ARE CORRECT
C     IF KDELA.GT.MAX THE SET IS TRUNCATED AFTER MAX POINTS
      LENG=MAX
      RETURN
C
C     FINISHED NORMALLY
C     LENG IS EITHER ZERO, OR LESS THAN MAX (AND POSITIVE)
    5 LENG=J-1
      RETURN
      END

C
C     SGNABS RETURNS THE SIGN OF I IN J AND THE ABS VALUE IN K
C
      SUBROUTINE SGNABS(I,J,K)
      INTEGER I,J,K
      IF (I.LT.0) GOTO 1
      J=+1
      K=I
      RETURN
    1 J=-1
      K=-I
      RETURN
      END
```

References:

BRESENHAM, J. E. (1965). Algorithm for computer control of a digital plotter, *IBM Systems Journal*, Vol. 4, No. 1, pp. 25–30.

STOCKTON, F. G. (1963). Algorithm 162, XYMove Plotting, *CACM*, Vol. 6, No. 4, p. 161.